1. **Escribir un bloque PL/SQL que calcule la media de dos números dados por el usuario.**

DECLARE

v\_num1 NUMBER :=&a;

v\_num2 NUMBER :=&b;

v\_suma NUMBER :=0;

BEGIN

v\_suma:=(v\_num1+v\_num2)/2;

DBMS\_OUTPUT.PUT\_LINE(v\_suma);

END;

1. **Escribir un bloque PL/SQL que pida dos identificadores de profesores y determine cual de los dos está más cerca de cero o si son iguales.**

DECLARE

v\_num1 NUMBER (8):=&a;

v\_num2 NUMBER(8) :=&b;

v\_num1abs NUMBER (8):=v\_num1;

v\_num2abs NUMBER (8):=v\_num2;

BEGIN

--Valores absolutos

IF v\_num1 < 0 THEN

v\_num1abs:=v\_num1\*(-1);

END IF;

IF v\_num2 < 0 THEN

v\_num2abs:=v\_num2\*(-1);

END IF;

--Comprobación

IF v\_num1abs=v\_num2abs THEN

DBMS\_OUTPUT.PUT\_LINE(v\_num1||' es igual que '||v\_num2);

ELSIF v\_num1abs<v\_num2abs THEN

DBMS\_OUTPUT.PUT\_LINE(v\_num1||' está más lejos de cero que '||v\_num2);

ELSIF v\_num1abs>v\_num2abs THEN

DBMS\_OUTPUT.PUT\_LINE(v\_num1||' está más cerca de cero que '||v\_num2);

END IF;

END;

1. **Escribir un bloque PL/SQL que convierta una determinada cantidad expresada en segundos a horas, minutos y segundos.**

--Incompleto

DECLARE

v\_s\_int INTEGER:=&a;

v\_s\_float NUMBER(8,4);

v\_m\_float NUMBER(8,4);

v\_h\_float NUMBER(8,4);

BEGIN

v\_s\_float:=mod(v\_s\_int,60);

v\_m\_float:=trunc(v\_s\_int/60);

v\_h\_float:=trunc(v\_m\_float/60);

DBMS\_OUTPUT.PUT\_LINE('seg: '||v\_s\_int);

DBMS\_OUTPUT.PUT\_LINE('s: '||v\_s\_float);

DBMS\_OUTPUT.PUT\_LINE('m: '||v\_m\_float);

DBMS\_OUTPUT.PUT\_LINE('h: '||v\_h\_float);

END;

1. **Mostrar el contenido de una variable que contenga la capacidad total de todas las aulas del edificio de Informática (EUI)  
     
   create table LOCALES (codigo varchar2(10) primary key, nombre varchar2(30),capacidad number(10), edificio varchar2(40));  
     
   insert into locales values('inf1','sexto',20,'informatica');  
   insert into locales values('inf2','primero',15,'informatica');  
   insert into locales values('inf3', 'segundo',20,'informatica');  
   insert into locales values('inf4', 'cuarto',19,'informatica');  
   insert into locales values('inf5', 'quinto',17,'informatica');  
   insert into locales values('fin1', 'primero',20,'finanzas');  
   insert into locales values('com1', 'cuarto',20,'comercial');  
   insert into locales values('elec1', 'primero',20,'electronica');**

DECLARE

v\_capacidad NUMBER(2);

BEGIN

SELECT sum(capacidad)

INTO v\_capacidad

FROM locales

WHERE lower(edificio)='informatica';

DBMS\_OUTPUT.PUT\_LINE('Suma total: '||v\_capacidad);

END;

1. **Mostrar el contenido de una variable que nos dé la suma de los salarios de FORD y de ADAMS**

DECLARE

v\_msal1 NUMBER(9,2);

v\_msal2 NUMBER(9,2);

v\_suma NUMBER(9,2);

BEGIN

SELECT msal

INTO v\_msal1

FROM employees

WHERE upper(ename)='FORD';

SELECT msal

INTO v\_msal2

FROM employees

WHERE upper(ename)='ADAMS';

v\_suma:=v\_msal1+v\_msal2;

DBMS\_OUTPUT.PUT\_LINE('FORD:'||v\_msal1);

DBMS\_OUTPUT.PUT\_LINE('ADAMS: '||v\_msal2);

DBMS\_OUTPUT.PUT\_LINE('TOTAL: '||v\_suma);

END;

1. **Escribir un procedimiento que reciba dos números y visualice su suma.**
2. **Codificar un procedimiento que reciba una cadena y la visualice al revés.**
3. **Escribir una función que reciba una fecha y devuelva el año, en número, correspondiente a esa fecha.**
4. **Escribir un bloque PL/SQL que haga uso de la función anterior**
5. **Dado el siguiente procedimiento:  
   CREATE OR REPLACE PROCEDURE crear\_depart (  
   v\_num\_dept depart.dept\_no%TYPE,  
   v\_dnombre depart.dnombre%TYPE DEFAULT 'PROVISIONAL',  
   v\_loc depart.loc%TYPE DEFAULT 'PROVISIONAL')  
   IS  
   BEGIN  
   INSERT INTO depart  
   VALUES (v\_num\_dept, v\_dnombre, v\_loc);  
   END crear\_depart;**

**Indicar cuáles de las siguientes llamadas son correctas y cuáles incorrectas, en este último caso escribir la llamada correcta usando la notación posicional (en los casos que se pueda):  
crear\_depart; -- 1º  
crear\_depart(50); -- 2º  
crear\_depart('COMPRAS'); -- 3º  
crear\_depart(50,'COMPRAS'); -- 4º  
crear\_depart('COMPRAS', 50); -- 5º  
crear\_depart('COMPRAS', 'VALENCIA'); -- 6º  
crear\_depart(50, 'COMPRAS', 'VALENCIA'); -- 7º  
crear\_depart('COMPRAS', 50, 'VALENCIA'); -- 8º  
crear\_depart('VALENCIA', 'COMPRAS'); -- 9º  
crear\_depart('VALENCIA', 50); -- 10º**

1. **Desarrollar una función que devuelva el número de años completos que hay entre dos fechas que se pasan como argumentos.**
2. **Escribir una función que, haciendo uso de la función anterior devuelva los trienios que hay entre dos fechas. (Un trienio son tres años completos).**
3. **Codificar un procedimiento que reciba una lista de hasta 5 números y visualice su suma.**
4. **Escribir una función que devuelva solamente caracteres alfabéticos sustituyendo cualquier otro carácter por blancos a partir de una cadena que se pasará en la llamada.**
5. **Implementar un procedimiento que reciba un importe y visualice el desglose del cambio en unidades monetarias de 1, 5, 10, 25, 50, 100, 200, 500, 1000, 2000, 5000 Ptas. en orden inverso al que aparecen aquí enumeradas.**
6. **Codificar un procedimiento que permita borrar un empleado cuyo número se pasará en la llamada.**
7. **Escribir un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la localidad nueva.**
8. **Crear un procedimiento que reciba como parámetros el número de un empleado y, por referencia, se le pase un parámetro de tipo localidad del departamento del empleado. En la llamada ese parámetro irá sin valor. El procedimiento recogerá la localidad del departamento si la tuviere. En un bloque anónimo, llamaremos al procedimiento y evaluaremos dicha localidad, si tiene valor, lo sacaremos por pantalla, si no tiene nada, diremos que está vacío.**
9. **Visualizar todos los procedimientos y funciones del usuario almacenados en la base de datos y su situación (valid o invalid).**